

Implementation of a Progressive Web App for Teacher Attendance and Payroll System at PAUD Mckids West Bekasi

Devara Isbani Yusuf ^{1*}, Veri Arinal ², Mesra Betty Yel ³

^{1*,2,3} Department of Informatics Engineering, Faculty of Computer Technology, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, East Jakarta City, Special Capital Region of Jakarta, Indonesia.

Article History:

Received: July 29, 2026.

Revised: July 30, 2026.

Accepted: August 27, 2026.

Available online: September 20, 2026.

Published: December 1, 2026.

*Corresponding Author:

devarayusuf454@gmail.com

Abstract: PAUD McKids West Bekasi is an early childhood education institution that previously managed teacher attendance using manual paper logbooks and calculated payroll via basic spreadsheets. Consequently, the monthly recapitulation process required 2–3 working days and remained susceptible to human calculation errors. This study aims to design and implement a Progressive Web App (PWA) for the teacher attendance and payroll system, named SIPAG McKids, providing a computerized, integrated, and multi-device accessible administrative platform. The research employed the Research and Development (R&D) method using the Waterfall software engineering model, encompassing requirements analysis, system design, implementation, testing, and maintenance. The system was developed using the CodeIgniter 3 framework under the Model-View-Controller (MVC) architecture, a MySQL relational database, a Bootstrap 5 interface, DOMPDF for automated pay slip generation, and a Progressive Web App layer driven by a Web App Manifest and Service Workers to facilitate installation and offline caching. System evaluation comprised Black Box Testing for functional validation, PWA component verification, security assessments referencing the OWASP Top 10 guidelines, and performance audits using Google Lighthouse. The implementation results demonstrated that all Black Box testing scenarios functioned as designed. Security mechanisms—including Bcrypt password hashing, CSRF protection, Query Builder parameterization against SQL Injection, and Role-Based Access Control (RBAC)—operated correctly. Furthermore, self-service attendance based on selfie photography and GPS coordinates, alongside a tiered payroll approval workflow (Draft–Approved–Paid), operated successfully with transactional records verified in the production database. Google Lighthouse audit scores achieved 85 for desktop performance, 99 for mobile performance, 100 for Best Practices, 91 for SEO, and 78 for Accessibility. The system effectively enhances administrative recapitulation efficiency, improves calculation accuracy for tardiness deductions, ensures payroll transparency for teachers, and provides seamless mobile access for the institution.

Keywords: Codeigniter; Early Childhood Education; Payroll System; Progressive Web App; Teacher Attendance.

1. Introduction

Advancements in information and communication technology have increasingly prompted educational institutions to transition from labor-intensive manual administration toward web-based information systems capable of delivering accurate, real-time data to support operational productivity and structured decision-making (Puji Ikawati & Arinal, 2021). However, early childhood education centers frequently encounter operational constraints characterized by limited information technology budgets, lack of dedicated on-site technical personnel, and diverse personal mobile devices used by staff. Under such conditions, developing platform-specific native applications often introduces prohibitive development and maintenance costs. A modern, efficient web architecture addressing these constraints is the Progressive Web App (PWA), which integrates Service Workers and a Web App Manifest to provide installability on home screens, dependable offline execution, dynamic caching, and responsive native-like user interfaces without requiring formal deployment through commercial mobile application stores (Osmani, 2015). At the institutional level, PAUD McKids West Bekasi represents a community-scale early childhood education institution whose daily attendance recording and human resource management have remained reliant on physical logbooks and detached spreadsheet files. Observational findings and stakeholder interviews revealed that teachers documented daily arrivals through signatures on paper, while salary calculations were executed independently at the end of each monthly cycle. Consequently, the administrative reconciliation process required 2–3 working days because attendance ledgers and payment variables had to be cross-verified and aggregated by hand. This disconnected procedure fostered several operational vulnerabilities, including calculation discrepancies, delayed salary disbursements, difficulties in retrieving multi-semester historical data, and risks of unauthorized attendance manipulation. Furthermore, the absence of an automated self-service portal restricted institutional transparency, leaving teachers unable to monitor their personal attendance tallies, tardiness deductions, or itemized pay slips without consulting administrative personnel directly.

To address similar administrative bottlenecks, recent empirical literature has investigated digital interventions across various educational and organizational environments. Research on elementary school attendance systems demonstrated that a PWA-based approach provides a robust alternative to conventional paper logbooks, ensuring operational stability even within environments characterized by intermittent or low-bandwidth internet connectivity (Ramadhan & Widyawan, 2025). Concurrently, an investigation into integrated enterprise payroll mechanisms confirmed that establishing direct synchronization between real-time attendance logs and payroll engines significantly eliminates computational errors, prevents human bias in penalty calculations, and accelerates executive financial reporting (Wardani & Hendrawan, 2025). Complementary database engineering research also emphasizes that rigorous relational normalization and structural inter-module integration are essential to allow single-pass payroll processing without redundant data entry or transactional conflicts (Abdillah, 2006). The distinct contribution of this research lies in unifying automated attendance validation—leveraging geofenced GPS coordinates alongside real-time camera selfie verification—with a structured, three-tier payroll governance workflow (Draft–Approved–Paid) inside an integrated Progressive Web App tailored for resource-constrained early childhood education centers. This unified architecture operates with a lightweight client-side footprint, enabling teachers to install and operate the portal directly from their smartphone browsers without complex setup procedures. Accordingly, this study aims to design, construct, and evaluate the SIPAG McKids Progressive Web App, examining its functional integrity, structural application security, and runtime optimization standards to establish an accessible administrative automation model for small-scale educational institutions in Indonesia.

2. Related Work

A Progressive Web App (PWA) represents a modern web development paradigm leveraging contemporary browser capabilities to deliver user experiences comparable to platform-specific native applications, a concept originally formalized by Google (Osmani, 2015). According to Google Developers (2023), an effective PWA must satisfy three baseline criteria: capability (accessing hardware features such as cameras, push notifications, and local storage), reliability (maintaining continuous usability under intermittent or absent network connectivity via Service Workers), and installability (allowing users to pin the application to their device home screen). The underlying architecture relies on three primary components: a Service Worker functioning as a programmable network proxy that intercepts requests and manages client-side caching; a Web App Manifest formatted as a JSON file defining application metadata, icons, display orientations, and theme styling; and mandatory HTTPS deployment to preserve the cryptographic integrity of network request interceptions. Attendance management serves as an operational governance mechanism designed to record attendance transactions, enforce institutional discipline, calculate compensatory remuneration, and generate verifiable periodic reports (Hasibuan, 2017). Conventional paper-based attendance registers exhibit inherent

vulnerabilities, such as susceptibility to unauthorized physical alteration, labor-intensive manual reconciliation, and an absence of real-time monitoring. Implementing location-aware services and facial recognition within a PWA framework significantly reinforces transaction integrity, demonstrating that combining geolocation tracking with identity verification yields higher audit validity than manual logbooks (Bachtiar *et al.*, 2025). A computerized payroll system executes systematic compensation calculations and disbursement schedules, thereby reducing human calculation errors, truncating administrative processing intervals, and establishing auditable transaction trails. In this study, net salary calculations are automated based on attendance metrics, formulated as follows:

$$\text{Net_Salary} = \text{Base_Salary} + \text{Total_Allowances} - \text{Total_Deductions} \quad (1)$$

Where tardiness deductions and absenteeism penalties are automatically computed from monthly attendance timestamps, reflecting integration principles established in geolocation-based payroll architectures (Ghufron *et al.*, 2024). The web application backend is engineered using CodeIgniter 3, an open-source PHP framework structured upon the Model-View-Controller (MVC) architectural pattern, selected for its minimal execution footprint, complete compatibility with PHP 7.4, mature documentation, and native security mechanisms, including Cross-Site Request Forgery (CSRF) protection, form validation libraries, and Cross-Site Scripting (XSS) filtering (Upton, 2020). The implementation leverages the MVC pattern to isolate business logic from presentation layers, system hooks for authentication middleware, and an integrated Query Builder utilizing prepared statements to neutralize SQL Injection vulnerabilities. The user interface utilizes Bootstrap 5.3 to establish responsive grid layouts and CSS custom properties without external dependencies on jQuery (Bootstrap Team, 2023). Automated pay slip generation is handled by DOMPDF, which converts structured HTML and CSS templates into standardized, printable PDF documents (DOMPDF Project, 2023). System security conforms to the OWASP Top 10 guidelines (OWASP, 2021), incorporating one-way Bcrypt password hashing, session tokens, strict input sanitization, and dynamic Role-Based Access Control (RBAC) to enforce privilege boundaries prior to controller execution, maintaining authorization flexibility without hardcoded logic modifications (Zain *et al.*, 2021).

A critical review of prior literature was conducted to map technical architectures and empirical methodologies across web-based attendance and payroll implementations. A structured synthesis of relevant studies and the positioning of this research is detailed in Table 1.

Table 1. Summary of Prior Studies and Research Positioning

No.	Study Focus	Author (Year)	Summary of Findings	Research Gap	Research Positioning
1	PWA-based elementary student attendance	Ramadhan & Widyawan (2025)	PWA successfully replaced paper logbooks and sustained operational uptime under limited connectivity.	Excluded human resource management and payroll calculation modules.	Integrates PWA-based attendance verification with a multi-tier payroll engine.
2	Real-time integrated attendance and payroll	Wardani & Hendrawan (2025)	Automated salary calculations based on attendance records using SDLC principles.	Lacked PWA implementation, preventing direct installation on mobile hardware.	Combines automated payroll computation with an installable, mobile-responsive PWA layer.
3	Geolocation and facial recognition attendance via PWA	Bachtiar <i>et al.</i> (2025)	Geolocation and facial validation significantly improved student attendance accuracy.	Focused solely on student monitoring; omitted compensation workflows.	Adapts selfie and GPS validation for educators while coupling records directly to payroll ledgers.
4	Geolocation-based employee attendance and payroll	Ghufron <i>et al.</i> (2024)	Geolocation validated physical presence while wages were automatically calculated.	Implemented within a commercial retail setting rather than an educational institution.	Customizes integrated attendance and compensation logic to the operational profile of early childhood schools.

5	Web-based teacher attendance via Waterfall	Zain <i>et al.</i> (2021)	Transitioned teacher attendance tracking from paper logs to an administrative web platform.	Lacked GPS coordinate validation and integrated financial calculation modules.	Incorporates GPS-based presence verification and couples data directly to salary ledgers.
6	Relational database design for payroll systems	Abdillah (2006)	Third normal form (3NF) relational normalization improved data accuracy and query efficiency.	Developed prior to modern mobile and progressive web application frameworks.	Applies 3NF normalization within an integrated web database coupled to PWA client caching.
7	Web-based inventory system using RAD	Puji Ikawati & Arinal (2021)	Computerized data workflows eliminated data entry redundancy and accelerated reporting cycles.	Investigated material inventory tracking rather than human resource workflows.	Applies automated reporting principles to institutional attendance and compensation management.

As summarized in Table 1, previous studies have generally examined attendance management and payroll calculations in isolation, or deployed PWA capabilities exclusively for single-purpose student monitoring. The novelty of this study lies in unifying self-service attendance tracking (combining selfie photography and geofenced GPS coordinates) with a multi-tier payroll approval pipeline (Draft–Approved–Paid) within a single, installable Progressive Web App engineered specifically for the resource-constrained operational environment of PAUD McKids West Bekasi.

3. Methodology

3.1 Research Design and Setting

This study adopted the Research and Development (R&D) methodology, defined as a research approach utilized to design specific products and empirically assess their operational efficacy (Sugiyono, 2019). The resulting technological artifact is a web-based administrative system with Progressive Web App architecture, named SIPAG McKids, engineered specifically to manage teacher attendance tracking and payroll processing at PAUD McKids West Bekasi. The research locus encompasses institutional administrative operations involving three distinct user roles: system administrator, school principal, and teaching staff.

3.2 Data Collection Techniques

Data acquisition was conducted through three complementary techniques. Direct observation was conducted over a two-week period, monitoring physical attendance signing and spreadsheet-based salary calculations to identify operational bottlenecks and points of administrative latency. Semi-structured interviews were conducted with the School Principal, administrative staff, and two teachers to gather system requirements regarding monitoring dashboards, approval hierarchies, and mobile self-service expectations. A literature review was performed across academic journals, official framework documentation (CodeIgniter 3, Bootstrap 5, DOMPDF), and technical specifications published by the World Wide Web Consortium (W3C) and Google Developers. The data collection framework is summarized in Table 2.

Table 2. Summary of Data Collection Techniques

No.	Technique	Data Source	Objective	Output
1	Observation	Manual attendance and payroll operations at PAUD McKids	Identify operational bottlenecks and existing procedural latency	Baseline workflow description
2	Interviews	School Principal, administrative personnel, two teachers	Capture monitoring parameters, approval hierarchies, and feature requirements	Functional and non-functional requirements catalog
3	Literature Review	Academic journals, technical documentation of CI3, Bootstrap 5, DOMPDF, W3C PWA standards	Establish theoretical foundations and technical benchmarks	Architectural framework and implementation guidelines

3.3 Software Development Life Cycle

The software development process adhered to the classic Waterfall Model (Pressman, 2014), selected due to the well-defined and stable operational requirements gathered during initial field investigations. The development lifecycle progressed systematically through five sequential phases: Requirements analysis commenced by analyzing daily attendance records and compensation structures, resulting in a formal Software Requirements Specification (SRS) document. System design translated these requirements into technical schematics, incorporating the MVC architecture, an entity relationship diagram (ERD) encompassing 12 relational database tables, Bootstrap responsive wireframes, a Web App Manifest, and Service Worker caching configurations. Implementation involved programming core application modules, including user authentication, monitoring dashboards, employee registries, selfie-GPS attendance verification, automatic payroll calculations, and DOMPDPF report generation. System testing encompassed functional validation via Black Box Testing, PWA compliance audits, manual OWASP-based security penetration tests, and planned User Acceptance Testing. Finally, maintenance focused on deploying the artifact to the institutional local server, conducting administrative operator onboarding, and addressing minor runtime adjustments.

3.4 System Requirements Specification

The requirements engineering process yielded twelve functional requirements and eight non-functional requirements. The functional requirements govern authentication, administrative registries, geolocation attendance, and tiered payroll approvals, as enumerated in Table 3. The non-functional requirements establish performance benchmarks, security standards, responsive layout adaptability, and PWA criteria, as detailed in Table 4.

Table 3. Functional Requirements Specification

Code	Functional Requirement Description
FR-01	Multi-role user authentication with Bcrypt password verification and secure session termination.
FR-02	Role-Based Access Control (RBAC) governing navigational menus and administrative privileges.
FR-03	Visual statistical dashboard tracking active personnel, daily attendance tallies, and monthly salary metrics using Chart.js.
FR-04	Master teacher data management supporting CRUD operations for identity records, employment status, and profile photos.
FR-05	Daily attendance recording capturing five operational statuses: Present, Late, Excused, Sick, and Absent.
FR-06	Automated tardiness detection calculating exact penalty minutes relative to institutional shift thresholds.
FR-07	Automated net salary calculation executing the formula: Net Salary = Base Salary + Allowances - Deductions.
FR-08	Tiered payroll approval workflow executing sequential states: Draft, Approved, and Paid.
FR-09	Automated generation of individual pay slips in downloadable PDF format utilizing DOMPDPF.
FR-10	Administrative filtering tools and tabular data export for monthly attendance and compensation ledgers.
FR-11	Progressive Web App client configuration supporting home-screen installation across desktop and mobile browsers.
FR-12	Client-side static asset caching via Service Worker to sustain application responsiveness under low connectivity.

Table 4. Non-Functional Requirements Specification

Code	Non-Functional Requirement Parameter
NFR-01	Security: Bcrypt hashing, CSRF token validation, and parameter binding to neutralize SQL Injection.
NFR-02	Performance: Initial dashboard paint within two seconds, augmented by Service Worker static caching.
NFR-03	Responsiveness: Fluid user interface adaptation across desktop, laptop, tablet, and smartphone viewports.
NFR-04	PWA Compliance: Valid JSON manifest, active Service Worker lifecycle, and execution over HTTPS or localhost.

NFR-05	Compatibility: Seamless rendering across Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari Mobile.
NFR-06	Scalability: Capacity to manage records for up to 100 educators and 3,000 annual attendance transactions without latency.
NFR-07	Usability: Accessible typographic hierarchy, intuitive visual controls, and mobile-friendly touch targets.
NFR-08	Availability: High local server operational availability during institutional working hours.

3.5 System Architecture and Database Design

The structural design of SIPAG McKids integrates three architectural layers. The application framework employs CodeIgniter 3 under the MVC paradigm, wherein controllers process incoming HTTP requests, invoke data operations via models connected to MySQL, and render formatted views generated through Bootstrap 5.3. When the Service Worker is activated on client hardware, static runtime requests are intercepted at the browser level and served directly from Cache Storage, eliminating network latency. The relational database schema is normalized to the Third Normal Form (3NF) to eliminate data redundancy and preserve referential integrity. The schema comprises twelve interconnected tables linked via foreign key constraints: roles, menus, role_permissions, status_kepegawaian, jabatan, guru, users, absensi, penggajian, tunjangan, potongan, and ci_sessions. The client-side Progressive Web App configuration is established through three key components: a manifest.json file specifying application naming, a #C0392B theme palette, standalone display mode, and high-resolution icons (192×192 px and 512×512 px); an asynchronous Service Worker implementing a Cache-First strategy for static CSS, JavaScript, and font assets alongside a Network-First strategy for dynamic database transactions; and HTML meta tags enforcing consistent mobile browser chrome tinting.

3.6 Testing Procedures and Implementation Environment

System verification of SIPAG McKids was executed through three technical evaluation methods. Black Box Testing was performed across all twelve functional criteria (FR-01 through FR-12) to ensure boundary validity without inspecting underlying code routines. PWA compliance and runtime performance were evaluated using Google Chrome DevTools Lighthouse, auditing manifest validity, Service Worker registration, installability, secure execution context, and caching performance. Security assessments were executed manually referencing OWASP Top 10 standards, verifying Bcrypt entropy, CSRF protection, SQL Injection parameter handling, XSS output filtering, and RBAC authorization enforcement. Additionally, a User Acceptance Testing (UAT) framework was designed based on a Likert questionnaire measuring usability, functional alignment, transaction speed, interface legibility, and computational trust for prospective full deployment. The software and hardware execution environment is specified in Table 5.

Table 5 System Implementation and Evaluation Environment

No.	Architectural Layer	Technical Specification
1	Backend Framework	CodeIgniter 3 (PHP, MVC Pattern)
2	Relational Database	MySQL 8.0, InnoDB Storage Engine, 12 Normalized 3NF Tables
3	Frontend Presentation	Bootstrap 5.3, Poppins Typography
4	PDF Rendering Engine	DOMPDF Library
5	PWA Client Layer	Web App Manifest, Service Worker (Cache-First and Network-First Strategies)
6	Local Server Execution	Apache Web Server (XAMPP Environment), On-Premises School Network
7	Audit Tool	Google Chrome DevTools Lighthouse
8	Functional Testing	Black Box Testing (12 Test Scenarios, FR-01 through FR-12)
9	Security Verification	Manual Assessment Referencing OWASP Top 10 (Bcrypt, CSRF, SQLi, XSS, RBAC)

4. Results and Discussion

4.1 Results

The SIPAG McKids administrative portal was engineered using the CodeIgniter 3 framework adhering to the Model-View-Controller (MVC) architectural pattern, supported by a MySQL relational database named sipag, a Bootstrap 5 frontend interface, and a client-side Progressive Web App (PWA) runtime layer. The graphical interface adopts a responsive sidebar-content layout that dynamically renders menu items and

functional controls according to user privileges established in the menus and role_permissions tables. The implemented functional modules comprise:

- 1) A centralized Login page serving as the unified authentication gateway across all three administrative roles;
- 2) An operational Dashboard presenting analytical summary cards—tracking active teachers, daily attendance figures, and monthly salary commitments—supplemented by Chart.js visualizations illustrating weekly attendance distributions and aggregate payroll expenditure trends;
- 3) Teacher Data Management alongside Master Position and Employment Status modules governing core institutional personnel records;
- 4) Attendance Audit and Monthly Recapitulation modules;
- 5) A dedicated Self-Service Attendance page enabling teachers to execute location-validated check-ins using real-time selfie photography and device GPS coordinates, supported by manual exception handling when geolocation services are inaccessible;
- 6) Automated Payroll Generation and Payroll Ledger modules managing the sequential Draft–Approved–Paid approval lifecycle;
- 7) Individual pay slip PDF rendering powered by DOMPDF; and
- 8) A System Configuration suite incorporating User Account Management, Role and Permission assignments, and Menu Navigation controls.

The deployment infrastructure utilizes the institutional on-premises local server environment detailed in Table 5. By hosting the web services on an Apache server (XAMPP environment) within the school local network, PAUD McKids West Bekasi eliminated recurring cloud hosting expenses while retaining full data sovereignty over institutional human resource and financial records.

The sipag relational database was deployed on MySQL utilizing the InnoDB storage engine across all physical tables to enforce foreign key constraints, index lookups, and ACID transactional integrity (Oracle, 2023). The normalized database structure encompasses twelve relational tables: jabatan, guru, status_kepegawaian, users, roles, menus, role_permissions, absensi, penggajian, tunjangan, potongan, and ci_sessions. The jabatan table maintains baseline compensation parameters, including default base salaries, standard monthly allowances, and daily late-penalty deduction rates associated with each institutional role. The absensi table logs arrival and departure timestamps, categorical presence statuses, accrued tardiness minutes, and input methods distinguishing between physical administrative entry, checkin_app, or manual administrative corrections. For records generated via checkin_app, the table records latitude, longitude, and stored image paths capturing the employee selfie. The penggajian table stores pay slip master records alongside sequential workflow status indicators. Allowance and deduction line items are isolated within separate tunjangan and potongan tables under one-to-many relationships, permitting dynamic assignment of discretionary bonuses or institutional penalties to individual pay slips.

Black Box Testing conducted across all twelve functional test cases (FR-01 through FR-12) demonstrated full functional conformity with system specifications. Authentication routines, role-based view filtering, personnel data management, self-service selfie-GPS attendance logging, automated deduction calculations, multi-stage payroll approvals, and PDF pay slip rendering all functioned successfully. Empirical verification from the production database confirmed active transactional integrity, evidenced by persistent checkin_app records containing validated GPS coordinate pairs and image paths, alongside completed salary records advancing from Draft through Approved and Paid states. Application security assessments verified that user credentials were cryptographically hashed using one-way Bcrypt producing 60-character strings. CodeIgniter 3 built-in Query Builder parameterization successfully neutralized SQL Injection attack payloads on input fields. Automated CSRF token validation rejected unauthorized state-changing POST requests lacking valid tokens, and RBAC authorization filters systematically barred unauthorized direct-URL access attempts across privileged controller routes. Runtime performance and PWA compliance were evaluated using Google Chrome DevTools Lighthouse across desktop and mobile viewports. The audit results yielded a Performance score of 85 on desktop and 99 on mobile, accompanied by 100 for Best Practices, 91 for Search Engine Optimization (SEO), and 78 for Accessibility across both device categories. These metrics confirm exceptional mobile performance optimization and robust security configurations. However, the desktop performance score indicates opportunities for further payload optimization, while the accessibility rating highlights the necessity of refining visual color contrasts and assistive aria-label attributes to support diverse end-user requirements.

4.2 Discussion

From an operational standpoint, the deployment of SIPAG McKids successfully transformed administrative governance at PAUD McKids West Bekasi by replacing disconnected physical logbooks and spreadsheet calculations with a unified relational database. Administrative time expenditure was reduced substantially; attendance reconciliation and initial payroll drafting are now aggregated automatically from stored transactional records, eliminating an administrative cycle that previously consumed 2–3 working days each

month. Data precision improved because tardiness penalties and net compensation variables are computed automatically using synchronized server timestamps rather than retrospective manual calculations. From an organizational role perspective, administrative operators benefit from a consolidated data management workspace, the School Principal obtains executive oversight dashboards alongside authoritative payroll approval controls, and teaching personnel gain operational autonomy through self-service attendance tracking and unmediated access to historical compensation slips. The Progressive Web App layer enhances mobile accessibility by allowing the application to be installed directly onto teacher smartphones and accelerating subsequent load times via Service Worker caching. Nonetheless, this study possesses several empirical limitations. The operational evaluation was conducted within a single institutional environment with a compact user cohort. Furthermore, the Lighthouse accessibility score of 78 warrants targeted remediation of interface contrast ratios and screen-reader semantics. Finally, financial disbursements remain bounded by offline bank transfers or cash payouts, as the system currently lacks direct integration with commercial banking application programming interfaces or electronic payment gateways.

5. Conclusion and Recommendations

The design, implementation, and empirical evaluation of the SIPAG McKids application demonstrate that Progressive Web App technology can be effectively deployed to transform administrative governance within small-scale educational institutions. The application was successfully engineered using the CodeIgniter 3 MVC architecture coupled to an installable PWA client layer via a Web App Manifest and Service Workers, facilitating home-screen installation across mobile devices and enabling reliable self-service attendance verification using selfie photography and GPS coordinates. Centralizing institutional records into a normalized 12-table MySQL database and presenting them through a Bootstrap 5 responsive interface resolved previous data fragmentation. Operationally, the automated calculation of tardiness penalties and base salary components, alongside a three-tier approval workflow (Draft–Approved–Paid), substantively accelerated monthly reconciliation cycles from 2–3 working days to an automated, real-time process. Functional evaluation via Black Box Testing confirmed that all primary functional requirements operated in strict conformity with system specifications. Application security assessments verified the operational integrity of Bcrypt password hashing, automated CSRF protection, SQL Injection neutralization via parameter binding, and Role-Based Access Control filters. Furthermore, Google Lighthouse audits established strong operational benchmarks, achieving 99 on mobile performance, 100 on Best Practices, and 91 on SEO. Nevertheless, several empirical constraints warrant recognition: comprehensive quantitative User Acceptance Testing (UAT) could not be fully finalized within the reporting period due to ongoing questionnaire collection, and the accessibility rating of 78 highlights the necessity of interface contrast remediation. Future research should prioritize conducting structured, large-scale User Acceptance Testing involving all administrators, school leadership, and teaching personnel to quantitatively evaluate behavioral adoption and perceived usability using standardized metrics such as the System Usability Scale (SUS). Subsequent technical iterations should expand load and penetration testing under higher concurrent traffic, simulate active cross-site scripting and injection attack vectors, and periodically audit PWA compliance to optimize client caching and accessibility. Additionally, the platform could be enhanced by integrating automated financial disbursement via commercial payment gateways or direct banking application programming interfaces, supplemented by automated push notification services to alert staff regarding attendance deadlines and newly approved compensation slips.

References

- Abdillah, L. A. (2006). Perancangan basisdata sistem informasi penggajian (studi kasus pada Universitas 'XYZ'). *Jurnal Ilmiah Matrik*, 8(2), 135–152.
- Bachtiar, Y. D., Habib, A., & Koesdijarto, R. (2025). Rancang bangun sistem informasi absensi berbasis lokasi dan face recognition dengan Progressive Web Apps (PWA) untuk menentukan siswa unggul menggunakan metode SAW di SMKN 2 Ngawi. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(2), 3535–3541. <https://doi.org/10.36040/jati.v9i2.13044>
- Bootstrap Team. (2023). *Bootstrap: The most popular HTML, CSS, and JS library in the world* [Computer software]. <https://getbootstrap.com/>
- DOMPDF Project. (2023). *Dompdf: PHP library for converting HTML to PDF* (Version 2.0.3) [Computer software]. GitHub. <https://github.com/dompdf/dompdf>

- Ghufron, M., Meimaharani, R., & Murti, A. C. (2024). Sistem informasi absensi dan penggajian karyawan berbasis website menggunakan metode geolocation pada Toko Ayuri Homedress. *Jurnal Pendidikan dan Teknologi Indonesia*, 4(11), 439–452. <https://doi.org/10.52436/1.jpti.463>
- Google Developers. (2023). *What are Progressive Web Apps?* <https://web.dev/explore/progressive-web-apps>
- Hasibuan, M. S. P. (2017). *Manajemen sumber daya manusia*. Bumi Aksara.
- Oracle. (2023). *MySQL 8.0 reference manual* [Technical documentation]. Oracle Corporation. <https://dev.mysql.com/doc/>
- Osmani, A. (2015). *Progressive web apps*. Google Developers.
- OWASP. (2021). *OWASP Top 10:2021 - The ten most critical web application security risks*. Open Web Application Security Project. <https://owasp.org/Top10/>
- Pressman, R. S. (2014). *Software engineering: A practitioner's approach*. McGraw-Hill Education.
- Puji Ikawati, A., & Arinal, V. (2021). Penerapan metode RAD dalam sistem persediaan barang berbasis web pada PT. Agree Progress International di Jakarta Barat. *Jurnal Sosial dan Teknologi*, 1(8), 875–886. <https://doi.org/10.36418/jurnalsostech.v1i8.175>
- Ramadhan, A. D., & Widyawan, T. I. (2025). Rancang bangun sistem absensi siswa sekolah dasar berbasis PWA (Studi kasus: SDN Uwung Jaya Tangerang). *IKRAITH-Informatika*, 9(3), 1–10. <https://doi.org/10.37817/ikraith-informatika.v9i3>
- Sugiyono. (2019). *Metode penelitian kuantitatif, kualitatif, dan R&D*. Alfabeta.
- Upton, D. (2020). *CodeIgniter 3 user guide* (Version 3.1.13) [Technical documentation]. CodeIgniter Foundation. <https://codeigniter.com/userguide3/>
- Wardani, D. C., & Hendrawan, J. (2025). Penerapan sistem informasi penggajian terintegrasi dengan absensi real-time berbasis web pada CV. The Three Points menggunakan metode SDLC. *Juktisi: Jurnal Komputer Teknologi Informasi Sistem Komputer*, 4(2), 442–450. <https://doi.org/10.62712/juktisi.v4i2.440>
- Zain, A. R., Priyaditama, F. J., & Hermawan, I. (2021). Perancangan sistem presensi guru berbasis web menggunakan metodologi waterfall. *Jurnal Multinetics*, 7(2), 145–153. <https://doi.org/10.32722/multinetics.v7i2.4277>